

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
22 February 2001 (22.02.2001)

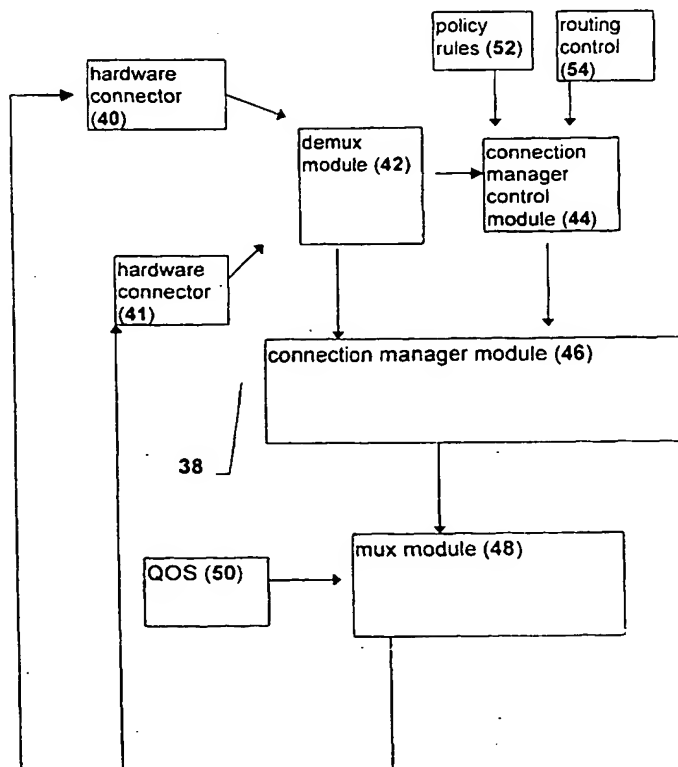
PCT

(10) International Publication Number
WO 01/13246 A1

- (51) International Patent Classification⁷: **G06F 13/00.** (74) Agent: RAVAD, Beit, Agish; Plinner, Bodner & Brass, 13 H04J 3/02, H04L 12/46, 12/66 Noach Mozes Street, 67442 Tel Aviv (IL).
- (21) International Application Number: PCT/IL00/00490 (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (22) International Filing Date: 11 August 2000 (11.08.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 09/372,875 12 August 1999 (12.08.1999) US (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (71) Applicant (*for all designated States except US*): EX-ALINK LTD. [IL/IL]; Business Park, 85 Medinat Hayehudim Street, 46766 Herzeliya (IL).
- (72) Inventor; and
- (75) Inventor/Applicant (*for US only*): ZAROM, Rony [IL/IL]; 9 Rabbi Akiva, 46423 Herzeliya (IL).
- Published:
— With international search report.

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR RAPID WIRELESS APPLICATION PROTOCOL TRANSLATION



(57) Abstract: A system for transmitting data units from a first type of network protocol to a second type of network protocol, including a network for transporting a unit of data and a translator gateway for receiving portion of the unit of data, which includes a demux module (42) for determining type of format for unit of data received and a connection manager (46) for determining if the portion of unit of data is sufficient to translate from the first type network protocol to the second type of network protocol and performing the translation if sufficient.

WO 01/13246 A1



-
- Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR RAPID WIRELESS APPLICATION PROTOCOL TRANSLATION

FIELD AND BACKGROUND OF THE INVENTION

5 The present invention relates to a system and method for rapid translation of data to and from the wireless application protocol (WAP) format, and in particular, for a system and method which enable real-time, "on the fly" translation through managing two separate, asynchronous sessions which are opened for communication both according to WAP and according to the current network protocols.

10 Cellular telephones are becoming increasingly popular for portable telephone use, particularly for users who are interested in rapid, mobile communication. As the amount of computational power and memory space which are available in such small, portable electronic devices becomes increased, a demand has arisen for different types of communication services through such devices. In particular, users have demanded that cellular telephones receive many
15 different types of multimedia data, including e-mail (electronic mail) messages and Web pages.

 In response to such demands, and to extend the power and efficacy of operation of portable, wireless electronic communication devices, the WAP (wireless application protocol) standard has been developed. WAP is now the standard for the presentation and delivery of wireless data, including multimedia and other information, and telephony services, on mobile
20 telephones and other types of wireless communication devices. WAP is designed to efficiently provide both multimedia and telephony services to such wireless communication devices, given the limitations of wireless networks and of the electronic devices themselves.

 Wireless communication devices have requirements and drawbacks which are different than cable-linked electronic devices. For example, wireless networks are frequently
25 significantly less stable than cable networks. Since users with such portable communication devices often operate these devices at different locations, the wireless network connection may not always be available, and may even suddenly become unavailable during a single communication session. In addition, the wireless communication devices themselves are more limited in terms of available resources than desktop computers. For example, such wireless
30 communication devices typically have a less powerful CPU (central processing unit), less memory, a lower amount of available power since these devices are often battery operated, and smaller display screens. Thus, wireless communication devices require adaptation to the

software and data transmission protocols in order to effectively deliver multimedia content from the Internet.

WAP provides the required adaptations and modifications to such software and data transmission protocols in order to meet the requirements of wireless communication devices.

5 For example, HTML (Hyper-text Mark-up Language) has been adapted to form WML (Wireless Mark-up Language). As explained in greater detail below, other adaptations and modifications have also been made, such that wireless communication devices require a translation system, or gateway, in order to communicate with computational devices being operated according to the current network protocols.

10 Background art Figure 1 is a schematic block diagram of currently available systems for providing connectivity between wireless communication devices operating according to WAP and computational devices being operated according to the current network protocols. A translation system 10 has a wireless communication device client 12, operating a WAP-compatible software agent 14. Requests are sent from wireless communication device client 12 to a proxy server 16. Proxy server 16 must receive an entire WAP-compatible request before translating into an original network protocol request, for example by translating WAP protocol instructions to HTTP and TCP/IP instructions. The translated request is then passed to an original server 18, which operates such software as CGI scripts 20 and which provides content 22.

20 The same process must also be followed in reverse when original server 18 serves the requested content. Proxy server 16 must receive the entirety of the requested content from original server 18 before translating the requested content into a WAP-compatible format. The translated content is then served to wireless communication device client 12 by proxy server 16.

25 In addition, proxy server 16 can only translate the data at the highest (application) level of the wireless or wired network protocols, which significantly decreases the efficiency of the translation process. The data must therefore be converted through all of the network layers before translation, and must then be reconverted to a format which is suitable for transmission through the physical network media, whether wired (cables) or wireless.

Clearly, background art system 10 has serious, significant disadvantages. First, two separate sessions are operated with significant delays in each session for the translation process, as proxy server 16 waits for the translation process to be completed for each original server 18 and wireless communication device client 12 before the translated data can be transmitted.

other session. Thus, since proxy server 16 must translate the entirety of the received information before passing it to the recipient, the translation process introduces significant delays. Background art system 10 is clearly inefficient and time consuming.

A more useful solution would be able to pass translated information as soon as only a portion is translated, such that the waiting period is significantly reduced. Furthermore, such a solution would only provide a simple translating device, thereby decreasing the complexity of the gateway. Unfortunately, such a solution is not currently available.

There is thus a need for, and it would be useful to have, a system and a method for rapidly translating data being transmitted according to WAP into known network protocol data without requiring the operation of two separate, simultaneous sessions, and without waiting for the entirety of the data before beginning the process of translating the data, such that the data is efficiently translated to and from a WAP-compatible format in real-time, without delays.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, wherein:

FIG. 1 is a schematic block diagram of a system according to the background art;

FIG. 2 is an exemplary WAP network which includes the translator/filter of the present invention;

FIG. 3 is a schematic block diagram of an exemplary gateway translator according to the present invention; and

FIG. 4 is a schematic block diagram of an exemplary connection manager from the gateway translator of Figure 3.

SUMMARY OF THE INVENTION

The present invention is of a method and a system for translating between data transmitted according to the WAP network protocols and data transmitted according to known network protocols, other than WAP protocols, such as known network protocols which are currently used for transmission of data on wired networks. The system and method of the present invention enable the translation process to be performed as soon as a minimal portion of data has been received by the gateway translator. This minimal portion is determined according to rules, such as the type of received data and flags within the received data. Therefore, the

translation process is performed according to an atomic, state machine mechanism on the smallest translatable unit of received data, rather than by waiting for a complete set of steps to occur by each session before the process of translation is continued. Thus, the method and system of the present invention are much faster and efficient than those which are known in the background art.

According to the present invention, there is provided a system for translating a unit of data from a first type of network protocol to a second type of network protocol, the system comprising: (a) a network for transporting the unit of data; and (b) a translator gateway for receiving at least a portion of the unit of data, the translator gateway comprising: (i) a demux module for determining a type of a format for the at least a portion of the unit of data, such that the at least a portion of the unit of data is in a first format of the first type of network protocol; and (ii) a connection manager for determining if the at least a portion of the unit of data is sufficient to translate from the first type of network protocol to the second type of network protocol and for translating the at least a portion of the unit of data from the first type of network protocol to the second type of network protocol, such that the at least a portion of the unit of data is in a second format of the second type of network protocol.

According to another embodiment of the present invention, there is provided a method for translating a unit of data from a first type of network protocol to a second type of network protocol, the unit of data being transported on a network, the method comprising the steps of: (a) receiving at least a portion of the unit of data from the network; (b) determining a format for the at least a portion of the unit of data, such that the at least a portion of the unit of data is in a first format of the first type of network protocol; (c) analyzing the at least a portion of the unit of data to determine if the at least a portion of the unit of data is sufficient for performing the translation to the second type of network protocol; and (d) if the at least a portion of the unit of data is sufficient, performing the translation to the second type of network protocol to form a translated unit of data, thereby obviating a need for receiving the entirety of the unit of data before initiating the translation.

Hereinafter, the term "network" refers to a connection between any two electronic devices which permits the transmission of data.

Hereinafter, the term "wireless device" refers to any type of electronic device which permits data transmission through a wireless channel, for example through transmission of radio waves. Hereinafter, the term "cellular phone" is a wireless device designed for the transmission of voice data and/or other data, through a connection to the PSTN (public switched telephone

network) system.

Hereinafter, the term "computer" includes, but is not limited to, personal computers (PC) having an operating system such as DOS, Windows™, OS/2™ or Linux; Macintosh™ computers; computers having JAVA™-OS as the operating system; and graphical workstations such as the computers of Sun Microsystems™ and Silicon Graphics™, and other computers having some version of the UNIX operating system such as AIX™ or SOLARIS™ of Sun Microsystems™; or any other known and available operating system. Hereinafter, the term "Windows™" includes but is not limited to Windows95™, Windows 3.x™ in which "x" is an integer such as "1", Windows NT™, Windows98™, Windows CE™ and any upgraded versions of these operating systems by Microsoft Corp. (USA).

Hereinafter, the term "Web browser" refers to any software program which can display text, graphics, or both, from Web pages on World Wide Web sites. Hereinafter, the term "Web page" refers to any document written in a mark-up language including, but not limited to, HTML (hypertext mark-up language) or VRML (virtual reality modeling language), dynamic HTML, XML (extended mark-up language) or related computer languages thereof, as well as to any collection of such documents reachable through one specific Internet address or at one specific World Wide Web site, or any document obtainable through a particular URL (Uniform Resource Locator). Hereinafter, the term "Web site" refers to at least one Web page, and preferably a plurality of Web pages, virtually connected to form a coherent group.

Hereinafter, the term "applet" refers to a self-contained software module written in an applet language such as Java or constructed as an ActiveX™ control.

Hereinafter, the phrase "display a Web page" includes all actions necessary to render at least a portion of the information on the Web page available to the computer user. As such, the phrase includes, but is not limited to, the static visual display of static graphical information, the audible production of audio information, the animated visual display of animation and the visual display of video stream data.

The method of the present invention could be described as a series of steps performed by a data processor, and as such could optimally be implemented as software, hardware or firmware, or a combination thereof. For the present invention, a software application could be written in substantially any suitable programming language, which could easily be selected by one of ordinary skill in the art. The programming language chosen should be compatible with the computer hardware and operating system to which the software application is

executed. Examples of suitable programming languages include, but are not limited to, C, C++ and Java.

DETAILED DESCRIPTION OF THE INVENTION

5 The present invention is of a method and a system for translating data transmitted according to the WAP network protocols to, and from, data transmitted according to known network protocols. The system and method of the present invention enable the translation process to be performed by packet, as soon as a minimal portion of data has been received by the gateway translator. This minimal portion is determined according to the current state of the
10 translator, which may be affected by various factors, such as the type of received data and flags within the received data. Therefore, the translation process is performed in the lower protocol layers, rather than requiring the packet or a buffer of data to be transformed into higher layers according to an atomic, state machine mechanism only on the received data. Thus, the method and system of the present invention are much faster and efficient than those which are known in
15 the background art.

 Preferably, a plurality of task translators is created within each gateway translator, each being invoked for performing a specific translation task to translate a particular portion of data. Creating a plurality of such task translators enables a plurality of translation tasks to be performed simultaneously, as well as enabling a variety of different types of data to be
20 translated. More preferably, an atomic translation module performs the basic, low level of task of searching through a LUT (look-up table) for each part of the data to translate. The task translator then performs the process of translating specific layer 3 (application layer) data, including determining when the process of translation can begin. An associated state machine then manages the overall process of translation, in particular the optional but preferred process
25 for translating between TCP and WTP data and the process for packaging the translated data into a corresponding packet.

 In addition, it should be noted that the system and method of the present invention could also be more generally used for translating a unit of data transported on a network from a first type of network protocol to a second type of network protocol. Therefore, the present invention
30 is useful for many different types of translating between network protocols, rather than simply wired network protocols and WAP network protocols. However, for the purposes of clarity only and without intending to be limiting, the following description centers upon the translation of data between the known Internet (IP) network and WAP network protocols as an example.

of the present invention.

The principles and operation of a system and a method according to the present invention may be better understood with reference to the drawings and the accompanying description, it being understood that these drawings are given for illustrative purposes only and are not meant to be limiting.

Referring now to the drawings, Figure 2 is a schematic block diagram of an exemplary WAP network system 24, with regard to receiving WML (wireless mark-up language) content as the exemplary type of data to be transmitted. As shown, a wireless communication device 26 receives binary WML (WMLB) for displaying a Web page, for maximum data transmission in the limited bandwidth available. Wireless communication device 26 is connected to a wireless network 28. Wireless network 28 is connected to a WAP server 30 for receiving WML or other WAP content, directly produced as binary WML for example. Thus, wireless communication device 26 is able to communicate directly with WAP server 30 and to receive binary WML content directly from WAP server 30.

An HTTP Web server 32 is also shown, connected to a WAP proxy server 34. WAP proxy server 34 is connected to wireless network 28, and is also able to serve binary WML to wireless communication device 26. WAP proxy server 34 receives WML content, optionally directly from HTTP Web server 32, if it is capable of serving such content. Alternatively, WAP proxy server 34 receives WML content from an HTTP filter server 36, which translates HTTP content from HTTP Web server 32 to WML content for serving to WAP proxy server 34. HTTP filter server 36 can be implemented as a translator gateway according to the present invention, for greater efficiency of operation of WAP network system 24, as described below with regard to Figure 3.

Figure 3 is a schematic block diagram of the functional modules of a translator gateway according to the present invention. It should be emphasized that these functional modules could be implemented in a variety of different ways, including as software or as hardware, for example. It should also be emphasized that Figure 3 shows the overall functionality of the gateway and of the relationships between the different functional modules, rather than any absolute system architecture.

A gateway translator 38 receives packets from at least one known Internet (IP) wired network hardware connector 40 and at least one wireless communication network hardware connector 41, of which only one each is shown for the purposes of illustration only and without

intending to be limiting in any way. Optionally, wired hardware connector 40 could be combined with wireless communication network hardware connector 41. Packets received from wired network hardware connector 40 are passed to a demux module 42. Demux module 42 examines the IP header of the packet to determine the source and destination IP addresses of the packet. Packet type is preferably determined by examining the protocol field in the IP header in order to distinguish between the different types of packets.

Packets received by gateway translator 38 may be one of two types: regular IP packets: and WAP packets or other wireless network packets. Gateway translator 38 has the advantage that the translation process is performed entirely at the IP level, rather than requiring stripping of various layers until the data is translated at the application level, for example between HTML and WMLB. As described previously, currently available translators in the art require the data to be translated only at the highest (application) level of the network protocols, which significantly decreases the efficiency of these background art translators. Thus, gateway translator 38 is more efficient and is therefore able to translate packets more rapidly from protocol type to the other than background art translators.

Once the packet type, and source and destination IP addresses of the packet have been determined, demux module 42 passes this information to a connection manager control module 44. Connection manager control module 44 examines the information, and compares this information about the received packet to the information for existing sessions with other network elements. If the received information corresponds to information for an existing session, then the packet is considered to be part of that existing session and is passed to the corresponding connection manager module 46. Alternatively, connection manager control module 44 invokes a new connection manager module 46 if the packet belongs to a new session, for which such a connection manager module 46 has not yet been invoked.

As described in greater detail with regard to Figure 4 below, connection manager module 46 performs the translation of the packet into the corresponding network protocol as necessary, and then repackages the packet for transmission on the network. For example, if connection manager module 46 receives a regular IP packet for translation into a WAP network packet, then connection manager module 46 both performs the translation and repackages the translated packet data into a suitable format for a WAP network packet.

Connection manager module 46 then passes the repackaged packet data to a mux module 48, for transmission on the network. Mux module 48 is shown as being connected to

both wired network hardware connector 40 and wireless communication network hardware connector 41 in order to demonstrate that the process of receiving, translating and transmitting is actually circular, in that both the wired and wireless networks are able to communicate to each other by sending and receiving data through translation gateway 38.

5 Mux module 48 preferably queues the prepared packets for transmission on the correct network according to a priority rating. For example, certain types of packets could be given higher priority according to a criterion which is measured by mux module 48. These packets could then be given a higher priority for a more rapid transmission on the respective network. More preferably, mux module 48 determines the priority rating according to at least one quality
10 of service rule which is received from a quality of service (QOS) module 50.

According to a preferred embodiment of connection manager control module 44, each packet is also examined according to a policy rule in order to determine if the translation process should be performed. Such a policy rule could include a security rule for determining if the packet is allowed to enter translation gateway 38, and hence to pass through translation
15 gateway 38 to the other type of network. More preferably, connection manager control module 44 receives the policy rule or rules from a policy rules module 52. In addition, control manager control module 44 also more preferably receives routing information from a routing control module 54, in order to pass any necessary routing data to connection manager 46 for the preparation of the repackaged packet data.

20 Figure 4 is a schematic block diagram of an exemplary embodiment of connection manager 46 in more detail. Connection manager 46 controls the overall process of translation. First, connection manager 46 receives each packet from demux module 42 or connection manager control module 44. As described previously with regard to Figure 3, connection manager 46 can receive two different types of packets: a regular IP network protocol packet, or
25 a WAP (wireless) protocol packet.

A brief explanation of the WAP network layers is provided in order to explain the differences and similarities between TCP and WTP packets (see for example *WAP Architecture Specification*, version 30-April-1998, incorporated herein by reference, which is available at <http://www.wapforum.org> as of July 9, 1999, for a more detailed discussion of these network
30 layers). WTP operates in conjunction with UDP (user datagram protocol) services as a light-weight transaction oriented protocol, which is suitable for wireless networks. WTP is able to provide reliable datagram services without unnecessary information, such as data related to

problems of packet order, which are only experienced in wired and not wireless networks. Furthermore, WTP provides error handling services and other transaction-related services.

As connection manager 46 receives each packet from demux module 42 or connection manager control module 44, connection manager 46 also receives the identification of the type of each packet from demux module 42. Connection manager 46 then passes each packet to the proper state machine according to the packet type. For example, if the packet is a TCP packet, the packet is passed to a TCP state machine 56. Alternatively, if the packet is a WAP packet, the packet is passed to a WAP state machine 58. The following description centers upon the translation of TCP packets to WTP packets, it being understood that a similar process is followed for the reverse type of translation.

TCP state machine 56 first removes the IP header from the TCP packet. TCP state machine 56 also examines the IP header in order to determine the type of data contained within the packet. Next, TCP state machine 56 passes the packet to a translator task module 57, according to the type of data contained within the packet.

Translator task module 57 then examines the data in order to determine if sufficient information has been received in order to begin the process of translation. Generally, there are three possible situations in this regard: the entire packet may be translatable; only part of the packet may be translatable; or none of the packet may be translatable. Each situation depends upon the data contained within the packet. For example, when HTTP content is being sent in a packet or a plurality of packets, translator task module 57 would examine the HTTP header, which is typically contained in the first TCP packet, and is usually contained in a single packet. Subsequent packets, containing the HTTP commands themselves, may not be translatable as individual packets, such that a plurality of such packets must be received before the process of translation can begin.

In order to differentiate between these possibilities, translator task module 57 preferably features a TCP parser 60 for parsing the data in the packet. TCP parser 60 more preferably features a language tree, such that each part of the data is read as TCP parser 60 descends down the tree. TCP parser 60 most preferably continues reading data and traveling through the tree until a leaf is reached. At this point, translator task module 57 is able to determine that the smallest unit of translatable data for this packet and/or concatenated packets has been located.

If none of the packet is translatable, then translator task module 57 must wait for another packet or packets to arrive before the process of translation is continued. For example, actual

HTTP commands are typically sent in a plurality of packets, such that translator task module 57 preferably analyzes the received data with a sliding window. As soon as sufficient HTTP command data has been received for translation, translator task module 57 sends that portion of the HTTP command data to be translated to WSP data. The window is then moved such that the next portion of HTTP command data is examined and translated. A similar process may be performed in order to translate HTML data to WMLB data. Thus, translator task module 57 preferably enables the process of translation to begin as soon as sufficient data is received, which is determined according to the type of data being translated.

Alternatively, translator task module 57 separates that portion of the packet which is translatable, which as noted previously may include substantially the entirety of the packet. Translator task module 57 more preferably begins the process of translating the data in the TCP packet into the corresponding WAP format data as soon as sufficient data have been received. Therefore, translator task module 57 may begin the process of translation with a single TCP packet, unless a plurality of such packets is required in order to process the information.

Optionally and most preferably, translator task module 57 is able to begin the process of translation with a portion of a single TCP packet, such that translator task module 57 then concatenates these portions of packets to form an entire packet. Since TCP packets may optionally arrive out of order, translator task module 57 is preferably able to read the header in order to determine the correct packet order, for example when a plurality of TCP packets must be concatenated before the process of translation can begin.

Once the smallest translatable unit of data in the packet and/or packets has been located, translator task module 57 transfers this smallest translatable unit of data to a first buffer 62. A HTTP to WSP translator 64 reads the unit of data from first buffer 62 and translates it to the corresponding WSP data format according to a LUT (look-up table). Such a LUT is provided in the WAP specification, as incorporated by reference, and as such is not discussed in further detail herein.

HTTP to WSP translator 64 optionally and preferably handles the repetitive data translation tasks of looking for each portion of data in the LUT in order to determine the proper translation. Therefore, HTTP to WSP translator 64 preferably performs the simple tasks underlying the translation process, while translator task module 57 manages the translation task for each translatable unit of data. More preferably, HTTP to WSP translator 64 only translates specific layers of data, such as layer 5 (HTTP translations). Data

headers for other layers are most preferably flagged by TCP state machine 56, such that HTTP to WSP translator 64 preferably ignores this data during the translation process.

After HTTP to WSP translator 64 has translated the data, the translated data is optionally and preferably placed in a WAP queue 66. WAP state machine 58 then removes data from WAP queue 66 in chunks of a suitable size for packaging into WAP packets. For example, WAP state machine 58 preferably adds the necessary translated headers in order to package the data into a WTP packet, suitable for transmission on the wireless network. More preferably, the header data for each packet is translated separately, such that WAP state machine 58 is able to build a WAP header for the WAP packet according to a one-to-one correspondence of headers. WAP state machine 58 then passes the packet to mux module 48 for transmission (not shown). The WAP packet is then transmitted to the wireless communication device (not shown).

In addition, WAP state machine 58 optionally and preferably is able to "spoof" packets which do not contain any translated data, but which must be constructed and transmitted in order for the connection between gateway translator 38 and the data server, or other network element, to remain open, or to otherwise facilitate communication between gateway translator 38 and a network element. This creates a steady flow of packets which facilitates such communication, for example in order to prevent a "time out" from occurring.

Similarly, communication manager 46 also contains components for permitting WAP packets to be translated into TCP packets. In this situation, WAP state machine 58 first receives the packet and transmits the packet to a WAP task translator 67. Once sufficient data has been received in order to begin the process of translation, WAP task translator 67 passes the data to a second buffer 70. A WSP to HTTP translator 72 then receives the parsed data from second buffer 70, and translates the data as previously described, but in the opposite orientation (from WAP data, such as WSP, to TCP data, such as HTTP). The translated data is then passed to a TCP queue 74.

TCP state machine 56 receives the translated data, and prepares a TCP packet in a process which is similar to that described for preparing WAP packets. Again, preferably the headers are translated separately in order to permit the packet to be properly constructed as a TCP packet. TCP state machine 56 can also spoof TCP packets, substantially as previously described for WAP packets.

Also preferably, an additional WTP binary translator (not shown) is also able to translate WML data into WML binary data for transmission to the wireless communication device. As described previously, wireless communication devices preferably receive WML data in a binary format. WMLB, in order to increase the amount of data which can be translated in the limited
5 bandwidth which is available on the wireless network.

The system and method of the present invention clearly have a number of advantages over currently available background art methods. For example, if a proxy server is used for the translation process, two separate sessions are conducted such that each session must wait for the
10 other to complete a communication with the proxy server before the process of translation can begin. This is clearly inefficient and causes a significant amount of time to be wasted in simply waiting for the other session to complete. By contrast, in the system and method of the present invention, the two separate sessions are managed simultaneously and asynchronously, such that each session does not need to wait for the other session to complete before beginning the
15 process of translation. Furthermore, the system and method of the present invention bypass requirements for such time consuming steps as handshakes, and instead initiate a direct connection to the data server or other network element. Also, the present invention is able to analyze and translate data in its raw form, from the network as packets, while the background art requires such data to be concatenated into a buffer before translation. Thus, the system and
20 method of the present invention are clearly more efficient and are able to perform translation at a more rapid rate than currently known background art systems and methods.

It will be appreciated that the above descriptions are intended only to serve as examples, and that many other embodiments are possible within the spirit and the scope of the present
25 invention.

WHAT IS CLAIMED IS:

1. A system for translating a unit of data from a first type of network protocol to a second type of network protocol, the system comprising:
 - (a) a network for transporting the unit of data; and
 - (b) a translator gateway for receiving at least a portion of the unit of data, said translator gateway comprising:
 - (i) a demux module for determining a type of a format for said at least a portion of the unit of data, such that said at least a portion of the unit of data is in a first format of the first type of network protocol; and
 - (ii) a connection manager for determining if said at least a portion of the unit of data is sufficient to translate from the first type of network protocol to the second type of network protocol and for translating said at least a portion of the unit of data from the first type of network protocol to the second type of network protocol, such that said at least a portion of the unit of data is in a second format of the second type of network protocol.
2. The system of claim 1, wherein the unit of data is a packet featuring a header, and said demux module determines the first type of network protocol by analyzing said header.
3. The system of claim 2, wherein said connection manager further comprises:
 - (1) a first state machine for determining a type of data contained in the packet;
 - (2) a first parser for parsing said at least a portion of the unit of data to form parsed data and to determine that said at least a portion of the unit of data is sufficient to translate from the first type of network protocol to the second type of network protocol; and
 - (3) a first translator for translating said parsed data to form translated data.
4. The system of claim 3, wherein said connection manager further comprises:
 - (3) a second state machine for receiving said translated data and for packaging said translated data into a second unit of data in said second format of the second type of network protocol.

5. The system of claim 4, wherein said first translator features a LUT (look-up table) for holding each data equivalent in the first type of network protocol and the second type of network protocol

6. The system of claim 5, wherein the first type of network protocol is a wired network protocol and the second type of network protocol is a wireless network protocol.

7. The system of claim 6, wherein said wireless network protocol is WAP (wireless application protocol).

8. The system of claim 5, wherein said first translator gateway features a plurality of connection managers and said first translator gateway further comprises:

- (iii) a connection manager control module for examining the unit of data and for determining if a session for the unit of data is open, such that if said session is open, the unit of data is passed one of said plurality of connection managers corresponding to said session.

9. The system of claim 8, wherein said translator gateway further comprises:

- (iv) a policy rules module for holding at least one rule for determining if the unit of data is permitted to be translated.

10. The system of claim 9, wherein said translator gateway further comprises:

- (v) a mux module for receiving said second unit of data and for transmitting said second unit of data on said network.

11. The system of claim 10, wherein said translator gateway further comprises:

- (vi) a quality of service module for determining a priority rating for transmitting said second unit of data.

12. A method for translating data from a first type of network protocol to a second type of network protocol, the unit of data being transported on a network, the method comprising the steps of:

- (a) receiving at least a portion of the data from the network;

- (b) determining a format for said at least a portion of the unit of data, such that said at least a portion of the unit of data is in a first format of the first type of network protocol;
- (c) analyzing said at least a portion of the unit of data to determine if said at least a portion of the unit of data is sufficient for performing the translation to the second type of network protocol; and
- (d) if said at least a portion of the unit of data is sufficient, performing the translation to the second type of network protocol to form a translated unit of data, thereby obviating a need for receiving the entirety of the unit of data before initiating the translation.

13. The method of claim 12, wherein the unit of data is a packet featuring a header, such that step (b) includes the step of determining said format as the first type of network protocol by analyzing said header.

14. The method of claim 13, further comprising the steps of:

- (e) receiving at least a second portion of a unit of data from the network, said at least a second portion of the unit of data being in the second type of network protocol; and
- (f) translating said at least a second portion of the unit of data from the second type of network protocol to the first type of network protocol.

15. The method of claim 14, wherein the first type of network protocol is a wired network protocol and the second type of network protocol is a wireless network protocol.

16. The method of claim 15, wherein said wireless network protocol is WAP (wireless application protocol).

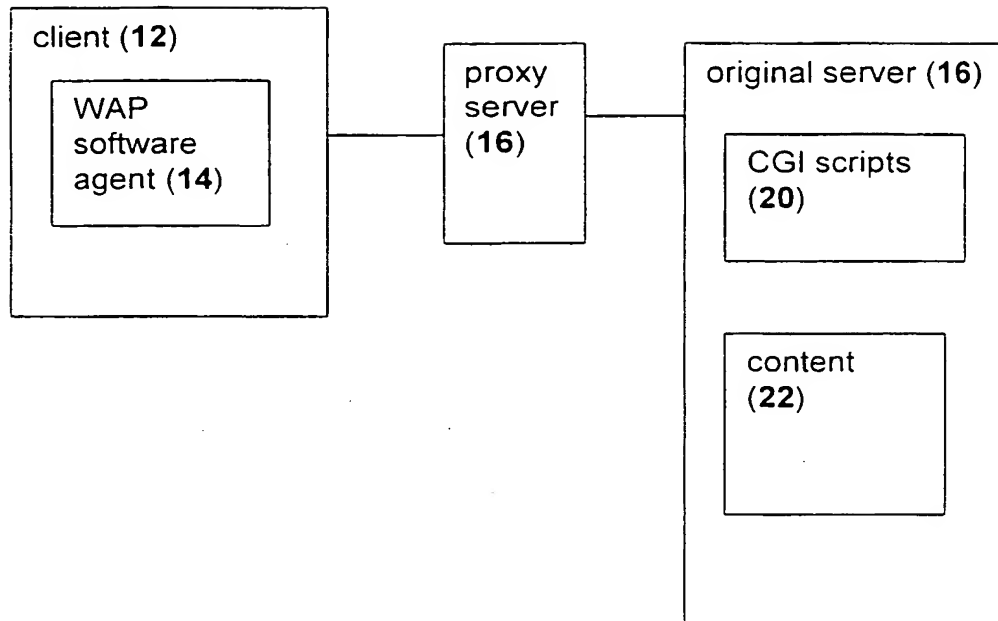
17. The method of claim 16, wherein step (d) includes the step of translating from WML (wireless mark-up language) data to HTML (hyper-text mark-up language).

18. The method of claim 16, wherein step (d) includes the step of packaging said translated unit of data as a complete unit of data according to the second type of network protocol.

19. The method of claim 14, further comprising the step of:
- (g) creating a spoofed packet to maintain a connection to the network.

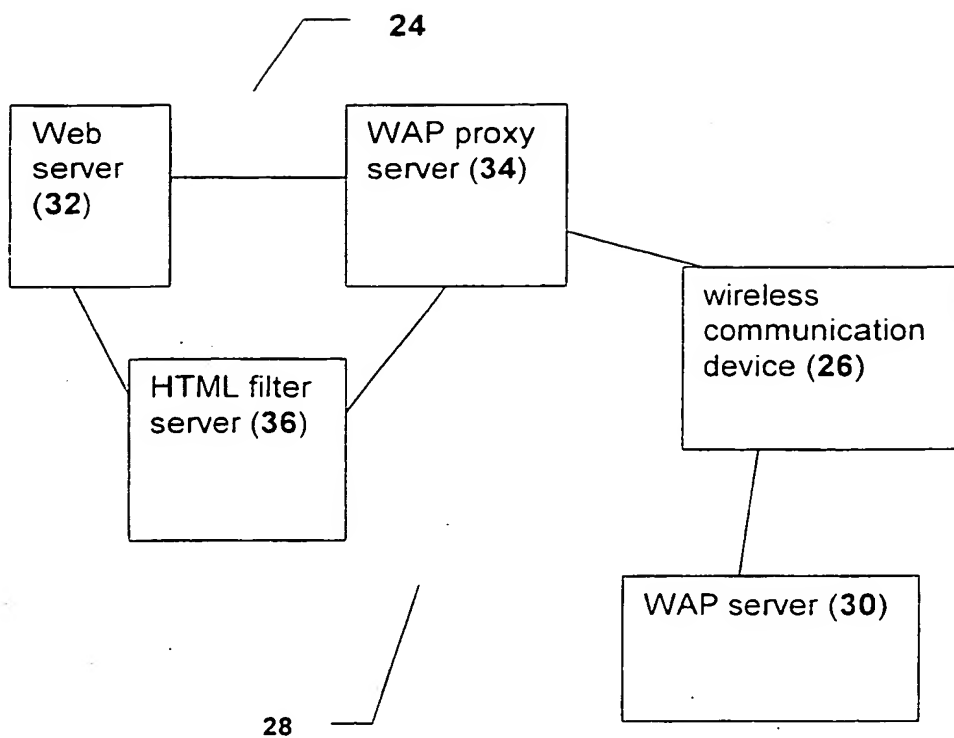
1/4

Background Art Figure 1



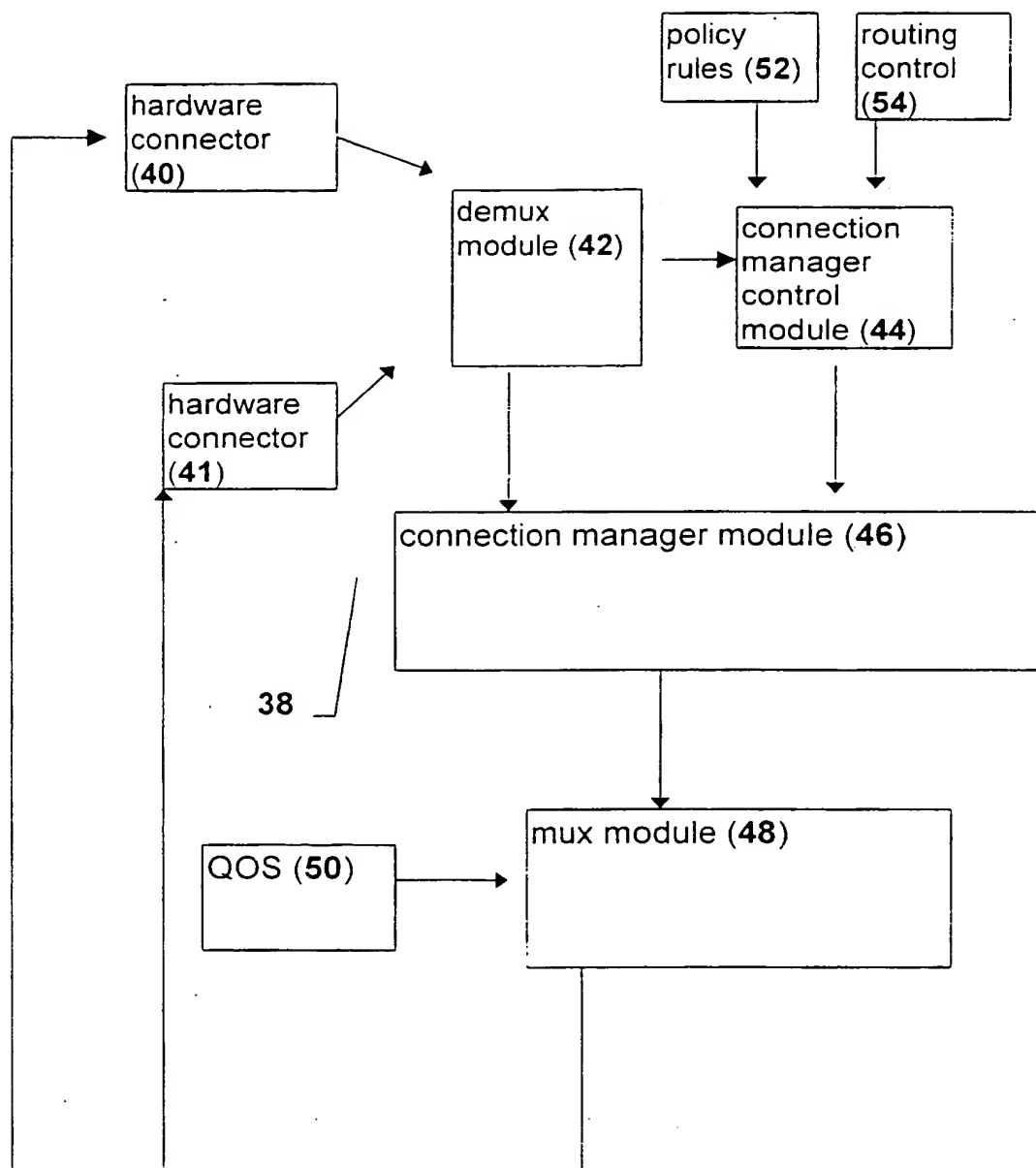
2/4

Figure 2



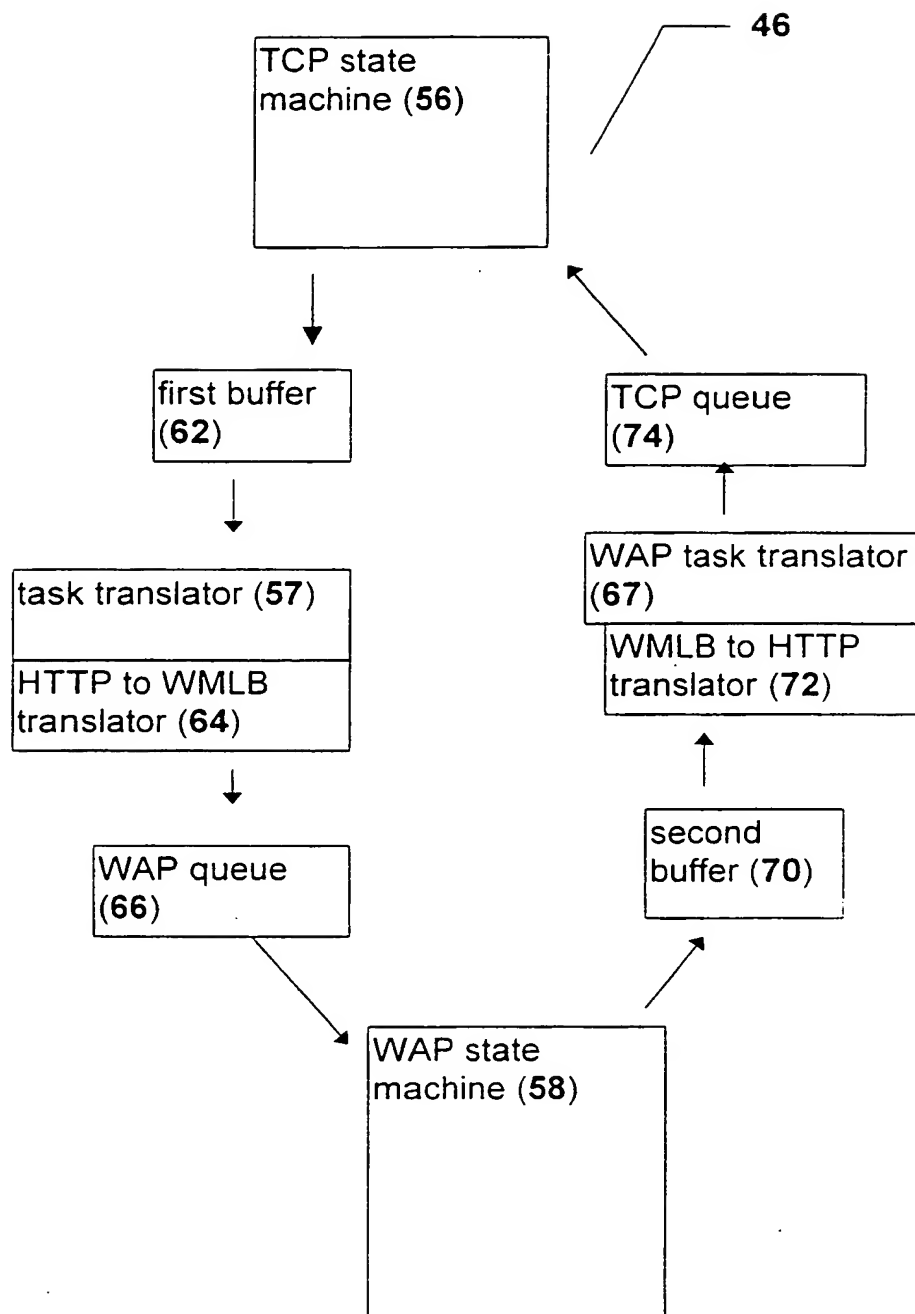
3/4

Figure 3



4/4

Figure 4



INTERNATIONAL SEARCH REPORT

International application No.

PCT/IL00/00490

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 13/00; H04J 3/02; H04L 12/46, 12/66
US CL : 370/231, 401, 466; 709/203, 230, 249

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/230, 231, 235, 352, 353, 354, 389, 392, 395, 400, 401, 465, 466, 467; 709/201, 203, 227, 230, 249; 455/517, 524

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Please See Continuation Sheet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,771,459 A (DEMERY et al) 23 June 1998, col. 1, line 30 to col. 2, line 58, col. 4, line 32 to col. 5, line 17.	1, 2, 6-16, 18, 19
---		3-5, 17
Y		1, 2, 6-16, 18, 19
X	US 5,894,478 A (BARZEGAR et al) 13 April 1999, col. 2, lines 38-50, col. 3, line 10 to col. 4, line 40.	3-5, 17
---		1, 2, 6-16, 18, 19
Y		3-5, 17
X, P	US 6,038,233 A (HAMAMOTO et al) 14 March 2000, col. 3, line 46 to col. 5, line 7, col. 6, line 19 to col. 7, line 5.	1, 2, 6-16, 18, 19
---		3-5, 17
Y, P		1, 2, 6-16, 18, 19
X, P	US 6,075,796 A (KATSEFF et al) 13 June 2000, col. 2, lines 19-52, col. 4, line 17 to col. 5, line 61.	3-5, 17
---		3-5
Y, P		1-19
Y	US 5,555,244 A (GUPTA et al) 10 September 1996, col. 4, line 19 to col. 5, line 21.	1-19
A	US 5,550,984 A (GELB) 27 August 1996, see entire document.	1-19
A	US 5,636,216 A (FOX et al) 03 June 1997, see entire document.	1-19
A	US 5,809,233 A (SHUR) 15 September 1998, see entire document.	

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such information being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

Date of mailing of the international search report

22 November 2000 (22.11.2000)

27 DEC 2000

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Authorized officer

Alpus H. Hsu

Telephone No. (703)305-4700

Facsimile No. (703)305-3230

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IL00/00490

Continuation of B. FIELDS SEARCHED Item 3: EAST

Search terms: packet, header, protocol, format, conver\$4 or translat\$4, state adj machine, pars\$3, wire, wireless, connection adj manager, policy

THIS PAGE BLANK (USPTO)